(19)日本国特許庁（ＪＰ）　　(12) **公開特許公報**（Ａ）　　(11)特許出願公開番号

特開平5－28280

(43)公開日　平成５年(1993)２月５日

(51)Int.Cl.⁵	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　15/72	４５０　Ａ	9192－5L		

審査請求　未請求　請求項の数４（全 13 頁）

(21)出願番号　特願平3－205539

(22)出願日　平成３年(1991)７月22日

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目１番６号

(72)発明者　成瀬　正
東京都千代田区内幸町一丁目１番６号　日本電信電話株式会社内

(72)発明者　新谷　幹夫
東京都千代田区内幸町一丁目１番６号　日本電信電話株式会社内

(72)発明者　斉藤　隆文
東京都千代田区内幸町一丁目１番６号　日本電信電話株式会社内

(74)代理人　弁理士　鈴木　誠

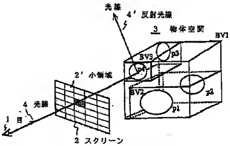
(54)【発明の名称】　光線追跡方法

(57)【要約】

【目的】　階層化された木構造の物体定義データを使って光線追跡し、画像を生成する際の高速な光線追跡処理の実現を図る。

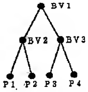
【構成】　画像生成対象の物体空間３について、図１の（ｂ）のように木構造の物体定義データが与えられる。この木構造に階層化された物体定義データを用い、画像面（スクリーン）２の小領域毎に、該小領域２′に投影される物体からなる部分木を動的に作成していく。そして、この部分木を利用して、画像面２の小領域毎に光線追跡を行い、該小領域の各画素の値を求め、画像を生成していく。この時、反射・屈折光については、更に部分木を作成し、同様に光線追跡を行う。これらの処理を各小領域について繰り返す。

（ａ）



本発明の光線追跡原理図

（ｂ）



物体の階層化構造

【特許請求の範囲】

【請求項１】　視点から物体の方向に向かって光線を追跡して物体の輝度を求めながら画像を生成する方法において、木構造に階層化された物体定義データを用い、画像面の小領域毎に、該小領域に投影される物体からなる部分木を動的に作成し、該部分木を利用して光線追跡を行うことを特徴とする光線追跡方法。

【請求項２】　追跡する光線に対して、部分木の各ノードに設定されたバウンディングボリュームを視点からの距離の近い順に動的に並べ換えて、該距離の近い順に部分木を探索することを特徴とする請求項１記載の光線追跡方法。

【請求項３】　小領域に投影される物体が同一の物体で占められるか否か判定し、同一の物体で占められる場合には該小領域内での部分木の探索を省略することを特徴とする請求項１あるいは２記載の光線追跡方法。

【請求項４】　反射・屈折光線については、小領域毎に、該小領域から反射・屈折する光線の大多数を包含する角錘あるいは円錐を作り、該角錐あるいは円錐に含まれる物体からなる部分木を動的に作成し、該部分木を利用して光線追跡を行うことを特徴とする請求項１、２あるいは３記載の光線追跡方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は光線追跡方法による画像生成の分野にかかり、詳しくは光線追跡法の高速処理方法に関する。

【０００２】

【従来の技術】光線追跡方法による画像生成法は、視点から物体の方向に向かって光線を追跡し、光線が物体と交差すれば、その点における物体の輝度を求めて画像面（スクリーン）に投影し、画像を生成する手法であるが、その問題点は計算にかなりの時間を要することである。この計算時間の大半は、光線と物体の交差判定に費やされる。そのため、交差判定回数の削減を狙ったデータ構造やアルゴリズムの提案が行われる。

【０００３】光線追跡アルゴリズムの高速化の本質は、つきつめれば、画像面の各画素に対して、その画素に投影される物体を効率よく求めることにある。それを目的として従来より種々の高速化技法が提案されてきた。従来の手法を分類すると、①データの階層化とＢoundind Volume（ＢＶ）の使用、②Ray coherencyの利用に大きく分けられる。

【０００４】物体定義データの階層化とＢＶの使用により交差判定回数の削減を図る方法はよく知られている。OctreeやConstructiveＳolid Geometry（ＣＳＧ）モデルを用いた階層化はその一例である。これらのモデルでは階層化された木の各ノードに対し、そのノードの子孫となる葉ノードの物体リミティブをすべて包含するようなＢＶを必要に応じて設定いる。これらのＢＶに対し

て交差判定を行うことにより交差計算の回数を大幅に削減できる。「ＢＶの体積を小さくするほど物体と交差する光線の数は少なくなる」という事実に基づいて、体積ができるだけ小さくなるＢＶを効率的に求める検討が行われている。

【０００５】一方、Arvoらは光線を視点と方向で分類し、類似した光線をグルーピングして、それらの光線と交差する物体の候補を求めることにより交差判定する物体の候補を求めることにより交差判定する物体の候補数を削減する手法を提案している（Arvo Ｊ．, Kirk Ｄ.:“Fast Ray Tracing by ray Ｃlassification”，Ｃomput. Graphics, Vol.２１. No.４, pp.５５－６４, Jul.１９８７）。この方法は、候補物体の分別が効率よくできるので、光線追跡法の高速化効果が大きい。

【０００６】

【発明が解決しようとする課題】上記いずれの方法も高速化効果は大きいが、さらに高速な処理を実現する必要がある。従来の手法は、階層化された木構造をそのまま使って光線追跡を行っていたために、少なくとも木の深さの交差判定計算が必要であり、これが、計算量削減の隘路となっていた。

【０００７】本発明の目的は、光線を追跡して物体の輝度を求めながら画像を生成する方法において、高速化効果が極めて高い光線追跡方法を提供することにある。

【０００８】

【課題を解決するための手段】上記目的を達成するために、請求項１は、視点から物体の方向に向かって光線を追跡して物体の輝度を求めながら画像を生成する方法において、木構造に階層化された物体定義データを用い、画像面の小領域毎に、該小領域に投影される物体からなる部分木を動的に作成し、該部分木を利用して光線追跡を行うことを特徴とする。

【０００９】請求項２の発明は、追跡する光線に対して、部分木の各ノードに設定されたバウンディングボリュームを視点からの距離の近い順に動的に並べ換えて、該距離の近い順に部分木を探索することを特徴とする。

【００１０】請求項３の発明は、小領域に投影される物体が同一の物体で占められるか否か判定し、同一の物体で占められる場合には該小領域内での部分木の探索を省略することを特徴とする。

【００１１】請求項４の発明は、反射・屈折光線については、小領域毎に、該小領域から反射・屈折する光線の大多数を包含する角錘あるいは円錐を作り、該角錐あるいは円錐に含まれる物体からなる部分木を動的に作成し、該部分木を利用して光線追跡を行うことを特徴とする。

【００１２】

【作用】本発明では、与えられた木構造の物体定義データから部分木を動的に作り、木の深さを浅くすることに

より、画像面（スクリーン）の小領域に投影される物体数は少なくなり、交差判定回数を削減することができる。さらに、部分木の各ノードレベルでの視点からの距離によるソーティング、小領域内の物体の一様性テストを加えることにより、光線と交差する候補物体の絞り込みが極めて効率よくでき、さらに高速処理が可能になる。また、反射・屈折光については、小領域毎に更に部分木を作り、該部分木を利用して光線追跡を行うことにより、１次光線のみならず反射・屈折光による光線追跡の画像も同時に生成される。

【００１３】

【実施例】以下、本発明の一実施例について図面により説明する。

【００１４】図１（ａ）は本発明の光線追跡方法の原理図を示している。図１において、１は目(視点)、２は画像面（スクリーン）、３は物体空間、４は光線である。画像生成対象の物体空間３上には多数の物体（プリミティブ）Ｐｉがある。この物体空間３を、各部分空間に含まれるプリミティブが１個になるまで階層的に分割し、各部分空間に対して、その空間に入るプリミティブをすべて包含するバウンディングボリューム（ＢＶ）を作り、物体の定義データを階層化する。図１（ａ）の物体空間３に対応する物体定義データの階層化構造を図１（ｂ）に示す。ここで、丸印はノードであり、バウンディングボリューム（ＢＶ）に対応するノードＢＶｉは、当該ＢＶの座標等を表わし、物体（プリミティブ）に対応するノードＰｉは、当該物体の方程式、表面属性、中心位置等を表わす。なお、この物体定義データの階層化については後述する。

【００１５】本発明では、画像面（スクリーン）２を多数の小領域２′に分割し、あらかじめ与えられた物体定義データの階層化構造を参照して、各小領域毎に交差判定を行い、該小領域に投影される物体からなる部分木を動的に作成する。そして、この部分木を利用して光線追跡を行い、各小領域毎に、該小領域に投影される物体の輝度を求め、画像面２にプロットする（レンダリング）。この時、反射光線（屈折光線）４′があれば、それに対する部分木も動的に生成し、該部分木を利用し更に光線追跡を行い、反射（屈折）物体の画像も該当小領域に重ねてプロットする。以上の処理を各小領域について繰返し実施する。

【００１６】図２及び図３に本発明の光線追跡方法による画像生成処理の一実施例のフローチャートを示し、図４に画像面（スクリーン）を小領域に分割する説明図を示す。

【００１７】図２は、木構造に階層化された物体定義データを用いて、画像面２の小領域２′毎に光線とＢＶあるいは物体（プリミティブ）との交差判定を行い、該小領域に対する部分木を作成するまでの処理である。本実施例では、処理は、画像面２の各小領域について、水平方向に左から右、垂直方向に上から下の順に行うとしている。図４は、作成された部分木を用いて光線追跡を行い（反射屈折光の追跡も行う）、一つの小領域について、それに投影されるプリミティブのレンダリングを行う部分である。該レンダリングを各小領域について繰返し行う。

【００１８】なお、図２及び図３の処理は、実際にはすべてコンピュータ上で実施されるものである。以下に、主な処理について詳述する。

【００１９】物体定義データの階層化

空間の分割法として、一様分割と、非一様分割がある。どちらを選択するかは、データ構造を作るコストと光線追跡のコストを総合して決めるべきであるが、一般にはデータに依存するので、善し悪しは一概にはいえない。ここでは、後者の分割を考える。後者の分割は木構造となる。具体的には空間をｋ個の部分空間に分割し、物体（プリミティブ）をそれぞれの部分空間に分ける。プリミティブが部分空間を跨ぐ場合は、いずれかの部分空間に入れる。たとえば、体積を最も大きく共有する部分空間に入れる。各部分空間が含むプリミティブが１個になるまでこの分割を再帰的に続ける。分割が終わったら、各部分空間に対し、その空間に入るプリミティブをすべて包含するバウンディングボリューム（ＢＶ）を作る。この分割により、ｋ進木のデータ構造ができる。ｋの値を決定する必要がある。

【００２０】木が平衡していれば次のように決めることができる。今、図５に示するようにｋ進木で階層化したとする。この木の各中間ノードにはＢＶが設定されている。この木を使って光線追跡すると、各光線に対して、まず、ルートノードのすべての子ノードに設定されたＢＶと交差判定が行われる。ついで、これらのノードのうち１つのノードが何らかの方法で選択され、そのすべての子ノードに対して交差判定が行われる。もし、バックトラックなしに、葉ノードまで交差判定が進めば、その光線に対する交差判定回数Ｎは、

【数１】

$$N = k \cdot \log_k n$$

となる。ここで、ｎはプリミティブ数である。

【００２１】ｎを固定したときＮを最小化するｋは、ｋ＝ｅ（自然対数の底）である。ｋは整数であるから実際にはｋ＝３がＮの最小値を与える。ついで、ｋ＝２，４となる。２と４は同じ値を与える。したがって、３進木あるいは２（４）進木構造とするのが有効である。すなわち、バックトラックが起こないか、あるいは発生が非常に少ないノード選択方法が与えられれば、３あるい２（４）進木による階層化が有利となる。以下では、他の高速化技法との親和性を考えて２進木を用いる。図１（ｂ）は２進木の簡単な例である。

【００２２】ＢＶの形状と交差判定

ＢＶはできるだけ体積が小さくなるように設定すること

が望ましいが、交差判定コストを考慮した設定をしなければならない。今、図６（ａ）に示すような各ノードにＢＶが設定された木構造を考え、その画像面（スクリーン）への投影が図６（ｂ）となったとする。ここで、投影面積をＳｉで表す。ノードｎ１に光線が当れば、その子ノードのＢＶとの交差判定が行われる。従って、ｎ１に光線が当たったという条件の下で、その子ノードとの交差判定に要する時間Ｔは、画面全体で、

【数２】

$$T = 2 C_1 S_1$$

となる。ここで、C_1はｎ１に設定されたＢＶとの交差判定コストである。

【００２３】バックトラックがなければサーチする木の各ノードに対してこの式が成立する。この場合には、この式から、単純な形状のＢＶを用いても交差判定コストが低ければ、トータルの計算時間は少なくなり得る。（投影面積が２倍になってもコストが半分なら計算時間は同じである。）以下では、ＢＶとして軸（世界座標）に沿った直方体を用いる。

【００２４】<u>動的部分木</u>

画像面（スクリーン）の各画素に対して、はじめに与えられた木を利用して光線追跡するのは、無駄が多い。画像面の小領域に投影されるプリミティブの数は限られているからである。そこで、与えられた木から小領域毎に動的に部分木を作り、小領域の中では部分木を利用して光線追跡をすることが考えられる。ここでは、１次光線（視点から追跡する光線）について述べる。小領域として、正方形領域あるいは長方形領域をとれば、部分木は、視点と領域の４隅の点から作られる角錐に入るかあるいは角錐と交差するプリミティブから作られる（この角錐内の光線の束を光束とよぶ）。この作成コストが高いと不利であるが、実際にはそれは低いことが次の考察からわかる。

【００２５】（イ）角錐作成

角錐は４つの平面から構成されるが、これらの平面を小領域毎に求める必要はない。画像面上で水平方向に並んだ小領域に対する角錐では、水平方向の面（上下２面ある）はいずれの角錐にも共通で同じ平面となる（例えば、図４のHF_0とHF_1など）。垂直方向に並んだ小領域では、垂直方向の面に対して同様の関係が成立する（図４のVF_0とVF_1など）。また、隣接する面は共有できる。平面の法線方向の半空間を外側、反対側を内側とすることにより、角錐の内部は一意に規定できる。従って、画像面の大きさをｎ×ｎ、小領域の大きさをｍ×ｍとすれば、全体で必要な平面の数は２（ｎ／ｍ＋１）となる。

【００２６】（ロ）動的部分木作成

木の各ノードに４つのフラグを与える。各フラグは、角錐の面に対応し、ノードに対応するＢＶあるいはプリミティブが面の内側にあるか、外側にあるか、あるいは交

差するかを示す。木のルートノードからはじめて各ノードのフラグを設定している。これは図２の交差判定で求まる。このとき、あるノードでの面の内側かあるいは外側になったらその子ノード以下の当該面に対するフラグ設定は不要となる。これは、ＢＶのもつ性質から明らかである。

【００２７】画像生成を小領域内、ついで水平方向の小領域の順に行うとすれば、水平方向に並んだすべての小領域に対して、水平面に対するフラグは共通に使用できる。また、垂直直面に対するフラグ設定は、水平面に対するフラグを参照することにより、不要な内外判定を省くことができる。図２は、このような処理を示している。これらのフラグを参照しながら部分木を構成する。すなわち、角錐と交差するかあるいは角錐に含まれるＢＶおよびプリミティブだけからなる部分木を構成する。このとき、図７に示す最適化も行う。作り方から明らかなように枝刈の効果が働くので、部分木作成コストは低く抑えることができる。なお、ここで述べた部分木作成法は２次光線（反射・屈折光）に対しても同じように適用できる。

【００２８】<u>ソーティング</u>

より視点に近いプリミティブが画像面に投影され、そのプリミティブを包含するＢＶは他のプリミティブを包含するＢＶと比べてもより視点に近い場合が多い、という事実に基づいて、部分木の各レベルで視点からの距離によるソーティングを行う。すなわち、あるノードのＢＶと光線が交差し、その子ノードに対して交差判定を行う必要が生じたら、子ノードを視点からの距離の近い順に並べ替える。これは、光線毎に行う。

【００２９】深さ方向優先の木探索を行う場合、木の各レベルでソーティングが行ってあれば、図８に示すように、ある子ノード以下を探索して得られたプリミティブの視点からの距離が次の子ノードに設定されたＢＶの視点からの距離より近いとき、その子ノード以下の探索は不要となる。このようにして、ソーティングを利用した枝刈ができる。

【００３０】２進木を用いれば、子ノードのソーティングは１回の比較だけで済み、極めて効率がよい。ここに２進木を用いた理由がある。

【００３１】<u>小領域内の一様性テスト</u>

小領域内が同一のプリミティブの投影像であることが何らかの方法でわかれば、この領域では、そのプリミティブのレンダリングを行えばよい。このことは、領域内で木の探索が不要となることを意味し、高速化に与える効果は大きい。また、このテストは、できる限り単純なものであることが要求される。

【００３２】ここでは、次のような簡単なテスト法を行う。小領域の４隅の点に対して光線追跡する。この段階で、部分木は、４番目に追跡した光線でソーティングされている。今、４隅の点のプリミティブが同一であった

とする。それをＰとする。このとき、小領域内でプリミティブＰより視点に近いところに他のプリミティブが無いことを調べればよい。部分木をＬeft-most Depth-Ｆirstでサーチし、プリミティブＱを見つける。Ｑは木の一番左端にある。木の構成のしかたから、ＱはＰより視点に近い。ＰとＱが同一でなければ、この領域は一様でないとする。ＰとＱが同一であったとする。このときＰの弟ノードを調べる。弟ノードはプリミティブかＢＶが設定されているかいずれかである。それをＲとする。Ｒが４番目に追跡した光線と交差するか調べる。これは、フラグをみるだけでよい。交差すれば、領域内は一様である。そうでなければ、一様でないとする。

【００３３】このテストは領域内で一様のときに必ず一様であるという判定を与えるものではないが、領域内が一様と判定されとき、プリミティブＰの前に他のプリミティブが無いことは保証される。図９、図１０にその例を示す。図９（ａ）は一様でないと判定される例、同（ｂ）は本来一様であるにもかかわらず、一様でないと判定される例である。図１０は一様と判定される例である。図１０の場合、プリミティブの前面に別のプリミティブはないが、小領域内にプリミティブでない部分があるので、その部分に対して部分木を用いて追跡をすることになる。なお、４隅にプリミディブが無いとき、領域内にプリミティブが無いための条件は部分木が空であることである。

【００３４】領域内が一様でないときは、領域を２分割して一様性テストをする。領域の大きさが閾値を下回らない範囲で、これを再帰的に繰り返す。

【００３５】<u>反射・屈折光束の追跡</u>

画面の矩形領域を通過する光束は角錐台で表現することが可能であり、これまでに述べてきた手法が適用できる。しかし、一般の反射・屈折光束を正確に記述することは困難である。ここでは、

① 反射・屈折光束を近似し、それを含む近似的な光束のバウンディングボリューム（以下、ＰＢＶと呼ぶ）を求める。

② 近似的なＰＢＶに対し、部分木を求め、一様性のテストを行う。

③ 各反射・屈折光線に対し、①で求めた近似ＰＢＶに含まれるか否を調べ、含まれる場合には②で得られた結果を利用する。

というアプローチをとる。ここで問題となるのは、いかにして近似ＰＢＶを求めるかという点である。

【００３６】光束の拡りが小さい場合には、近軸近似が反射・屈折の近似として有効であることが知られている（Ｓhinya M., Takahashi T., Naito S.：“Principesand Applications of Pencil Tracing”，Comput. Graphics, Vol.２１，No.４，pp.４５－５４，Jul. １９８７）。そこで、近軸近似を利用して近似ＰＢＶを求めることにする。

【００３７】近軸理論によれば、一点（たとえば視点）を発した近軸光束は２つの焦線を持ち、その焦線の方向は直交する。すなわち、近軸光束は図１１に示すように、互いに直交する交線を有する２つの平面の組に囲まれた領域として記述される。したがって、この領域を光束の近似として用いることは合理的である。しかし、実際の光束は理想的な近似光束ではなく、多くの光線が近似光束の領域外にはみ出すことになる（図１２（ａ））。そこで、光束の４隅の光線を基に、近似光束のＰＢＶを求めることを考える。この求め方は多数存在するが、簡単な方法として、４隅の光線の振れ角のうち最も大きな角をＰＢＶの拡り角とする手法がある。図１２（ａ）に示した２次元の例では、$|\theta 1| > |\theta 2|$であるので、図１２（ｂ）のようなＰＢＶが得られる。具体的には、以下の手順で近似光線ＰＢＶが求められる。

【００３８】① 画面小領域の４隅に対応する光線の反射・屈折光線ｉをＳnell則より求める。

② 得られた４本の光線の平均をとり、軸光線とする。

③ 軸光線をｚ－軸とする適当な光線座標ｘ－ｙ－ｚを求める。ただし原点は、４本の光線の始点の内、いちばん手前のものがｚ＝０となるようにとる。

④ ４本の光線の始点および方向を光線座標系に変換し、ｚ＝０の面との交点$r_i = (r_{xi}, r_{yi})$、および方向$S_i = (S_{ix}, S_{iy}) = ((dx/dz)_i, (dy/dz)_i)$を求める。

⑤ $r_{xmax} = \max(r_{xi})$, $r_{xmin} = \min(r_{xi})$, $r_{ymax} = \max(r_{yi})$, $r_{ymin} = \min(r_{yi})$および$s_{xmax} = \max(|s_{xi}|)$, $s_{ymax} = \max(|s_{yi}|)$を求める。

⑥ 求める４平面は次式で与えられる。

【数３】

$$x - s_{xmax} * Z = r_{xmax}$$

$$x + s_{xmax} * Z = r_{xmin}$$

$$y - s_{ymax} * Z = r_{ymax}$$

$$y + s_{ymax} * Z = r_{ymin}$$

なお、ｘ，ｙ軸のとり方は任意であるが、焦線方向にあわせてとるのが効率的である。

【００３９】この近似ＰＢＶ作成法は最も単純なものであり、さまざま変形が考えられる。その例として、以下のようなものが考えられる。

① ４隅の光線だけでなく、すべての画素に対応する光線の反射・屈折光線を求め、これらを同様の手法でバウンドすることも可能である。これにより追跡すべきすべての光線がＰＢＶに含まれることが保証される。

② このＰＢＶでは、光束が反射後に収束する場合は無駄な空間を含む（図１３（ａ））。したがって、収束光束については図１３（ｂ）に示すような２つの近束光束でバウンドする。

【0040】小領域が一様なら、4隅の点から光源を見込むＰＢＶにより部分木を作り、影のテストをする。平行光線なら、ＰＢＶは角柱となる。角柱を構成する各面の方線ベクトルは場所に依存せずどこでも一定となるので、角柱を作るコストは小さい。

【0041】次に、以下のアルゴリズムを用意し、本発明の高速化効果を示す。以下でAのアルゴリズムは2進木構造の使用を前提とする。

Ａ1　：ＢＶのみを使う。
Ａ2　：ＢＶとソーティングを使う。
Ａ2′：部分木とＢＶを使う。
Ａ3　：部分木とＢＶおよびソーティングを使う。
Ａ4　：全部を使う。（Ａ3＋一様性テスト）

また、比較の対象としてＡrvoの手法を用いる。

Ｂ1　：Ａrvoのアルゴリズム

【0042】次の5種類のデータについて高速化の効果＊＊の例を示す。

1）空間内に一定の半径の球を規則的に並べる。2通りの半径の球を用意する。（2種類）
2）空間内にランダムな半径を持つ球をランダムに配置する。（1種類）
3）1個および8個のティーポット（2種類）

【0043】画像生成例を図１４乃至図１８に示す。生成した画像の画素数は、５１２×５１２である。また、小領域は８×８の大きさを基本としている。図１４乃至図１６のプリミティブの総数は５１２であり、図１７、図１８のそれは、それぞれ５２２，４４１６である。ＢＶは軸に沿った直方体を使用している。

【0044】画像生成時間、交差判定回数を表1、表2に示す。

【表1】

計算時間

ｱﾙｺﾞﾘｽﾞﾑ	写真1	写真2	写真3	写真4	写真5
Ａ1	36'50"	20'46"	55'18"	12'19"	29'08"
Ａ2	14'07"	17'52"	21'18"	8'42"	18'52"
Ａ2'	5'12"	2'13"	7'16"	3'08"	7'56"
Ａ3	4'39"	2'17"	6'46"	2'29"	5'19"
Ａ4	3'45"	2'10"	5'22"	2'16"	5'03"
Ｂ1	6'55"	2'51"	9'34"	2'29"	6'06"

【表2】

交差判定回数

アルゴリズム		写真１	写真２	写真３	写真４	写真５
Ａ１	ＢＶ	57.9	37.9	86.6	20.4	50.9
	prim	19.8	6.88	28.8	4.61	9.60
	総数	20355	11744	30245	6552	15849
Ａ２	ＢＶ	20.2	29.7	30.9	13.5	30.5
	prim	3.75	5.01	7.28	2.53	4.69
	総数	6281	9087	10008	4190	9228
Ａ２'	ＢＶ	3.72	0.66	5.68	2.56	8.12
	prim	3.38	1.61	4.39	2.78	7.04
	総数	1859	595	2640	1400	3974
Ａ３	ＢＶ	3.16	0.65	5.09	1.86	4.88
	prim	2.27	1.56	3.48	1.60	3.42
	総数	1424	581	2248	908	2174
Ａ４	ＢＶ	1.93	0.65	3.18	1.75	4.81
	prim	2.04	1.79	2.99	1.52	3.20
	総数	1029	546	1608	893	2115
Ｂ１	prim	7.80	3.10	10.8	2.51	5.79
	総数	2045	813	2829	657	1516

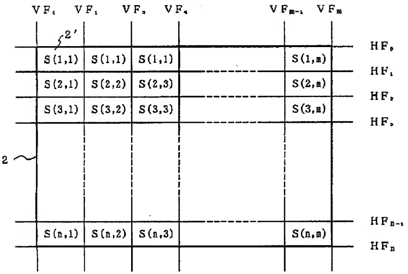
ＢＶとprimの項は光線１本あたりの交差判定回数（全光線に対する平均）
総数の項は画像面全体での総交差判定回数（ＢＶ＋プリミティブ）（×10³）

【００４５】この結果から、以下のように本発明のアルゴリズムの高速化効果がわかる。
（ａ） いずれのデータでもアルゴリズムＡ４が最も短時間で画像を生成する。また、Ａ３とＢ１を比較しても、Ａ３の画像生成時間がＢ１より少ない。
（ｂ） Ａ１とＡ２′の比較から部分木が高速化に寄与する効果が極めて高い。これは表２に示される如く、Ａ２′の交差判定回路が著しく減少することから得られる効果である。
（ｃ） 領域の一様性テストの効果は、プリミティブの大きさに依存する。図１５のように一様性領域の少ない場合でも、高速化効果がある。
（ｄ） Ａ２において、ＢＶとプリミティブの交差判定回路を比較するとわかるように、ソーティングを用いることにより、木探索時のバックトラックが極めて少なくことがわかる。

【００４６】
【発明の効果】以上説明したように、請求項１の発明では、与えられた木構造の物体定義データから景分木を動的に作り、木の深さを浅くすることにより、画像面（スクリーン）の小領域に投影される物体数は少なくなり、交差判定回数を削減することができる。
【００４７】請求項２及び３の発明では、部分木の各ノードレベルでの視点からの距離によりソーティング、小領域内の物体の一様性テストを加えることにより、光線と交差する候補物体の絞り込みが極めて効率よくでき、さらに高速処理が可能になる。
【００４８】請求項４の発明では、反射・屈折光については、小領域毎に更に部分木を作り、該部分木を利用して光線追跡を行うことにより、１次光線のみならず反射・屈折光による光線追跡の画像も同時に生成される。
【図面の簡単な説明】

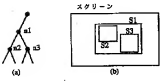
【図１】本発明の光線追跡方法の原理図である。
【図２】本発明の一実施例の処理フロー図である。
【図３】本発明の続きの処理フロー図である。
【図４】画像面（スクリーン）の分割を示す図である。
【図５】物体定義データの木構造の一般例を示す図である。
【図６】物体空間のバウンディングボリューム（ＢＶ）の投影面積と交差判定コストの関係を示す図である。
【図７】部分木の最適化を示す図である。
【図８】部分木のソーティングの説明図である。
【図９】小領域の一様性のテスト例を示す図である。
【図１０】同じく小領域の一様性のテスト例を示す図である。
【図１１】近軸光束の説明図である。
【図１２】光束のバウンディングボリューム（ＰＢＶ）の構成例を示す図である。　　＊
＊【図１３】ＰＢＶの改良例を示す図である。
【図１４】画像生成例である。
【図１５】画像生成例である。
【図１６】画像生成例である。
【図１７】画像生成例である。
【図１８】画像生成例である。
【符号の説明】
１　視点（目）
２　画像面（スクリーン）
２′　小領域
３　対象物体空間
４　光線
４′　反射光線
ＢＶ　バウンディングボリューム
Ｐ　物体（プリミティブ）

【図４】

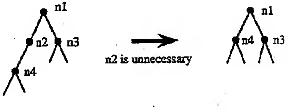


【図５】

【図６】

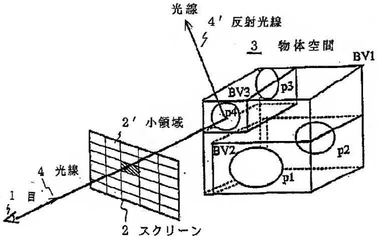


【図７】



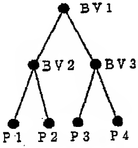
【図1】

（ａ）



本発明の光線追跡原理図

（ｂ）

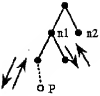


物体の階層化構造

【図18】

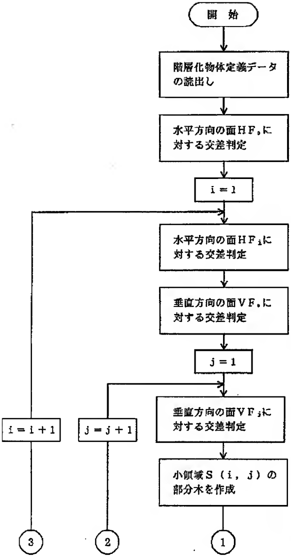
写真

【図8】

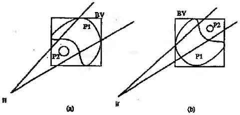


1. Searching descendants of node n1, get a primitive P closest to eye. Distance is t1.
2. Distance between BV set at n2 and eye is t2.
3. If t1<t2, it isn't necessary to search descendants of n2.

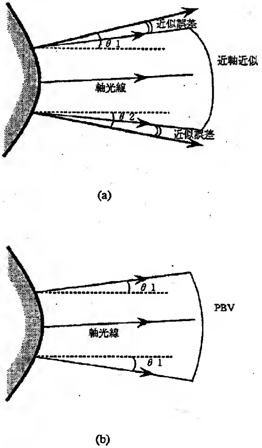
【図２】


開　始
階層化物体定義データの読出し
水平方向の面ＨＦ$_0$に対する交差判定
ｉ＝１
水平方向の面ＨＦ$_i$に対する交差判定
垂直方向の面ＶＦ$_0$に対する交差判定
ｊ＝１
垂直方向の面ＶＦ$_j$に対する交差判定
小領域Ｓ（ｉ，ｊ）の部分木を作成
ｉ＝ｉ＋１
ｊ＝ｊ＋１
3
2
1

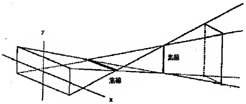
【図９】


BV
P1
P2
H
(a)
BV
P2
P1
H
(b)

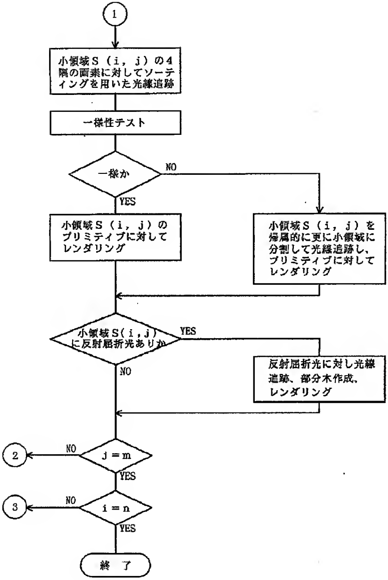
【図１２】


近似誤差
θ１
近軸近似
軸光線
θ２
近似誤差
(a)
θ１
PBV
軸光線
θ１
(b)

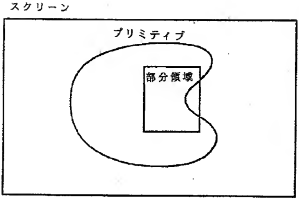
【図１１】


y
x

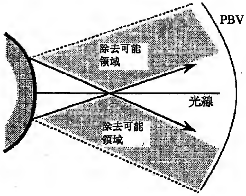
【図3】


1
小領域S（i，j）の4隅の画素に対してソーティングを用いた光線追跡
一様性テスト
一様か
NO
YES
小領域S（i，j）のプリミティブに対してレンダリング
小領域S（i，j）を帰属的に更に小領域に分割して光線追跡し、プリミティブに対してレンダリング
小領域S(i，j)に反射屈折光ありか
YES
NO
反射屈折光に対し光線追跡、部分木作成、レンダリング
NO
2
j＝m
YES
NO
3
i＝n
YES
終　了

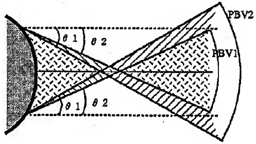
【図１０】



【図１３】



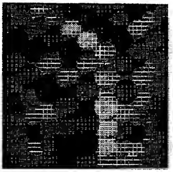
(a)



(b)

【図１４】

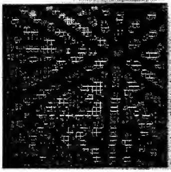
図面代用写真



写　真

【図１５】

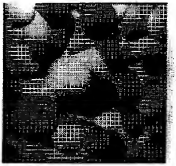
図面代用写真



写　真

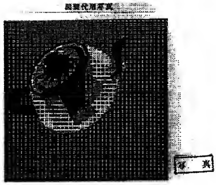
【図１６】

図面代用写真



写　真

【図17】



CLAIMS

[Claim(s)]

[Claim 1]In a method of generating a picture, while pursuing a beam of light toward the direction of objective from a viewpoint and asking for objective luminosity, A ray-tracing method creating dynamically subtree which consists of an object projected on this small region for every small region of an image face using object definition data hierarchized by tree structure, and performing ray tracing using this subtree.

[Claim 2]A ray-tracing method according to claim 1 rearranging dynamically into order with a near distance from a viewpoint bounding volume set as each node of subtree to a beam of light to pursue, and searching for subtree in order with this near distance.

[Claim 3]Claim 1 omitting search of subtree in this small region when an object projected on a small region judges whether it is occupied by the same object and occupied by the same object, or a ray-tracing method given in 2.

[Claim 4]About reflection and a refracted ray, **** or a cone which includes a large majority of a beam of light reflected and refracted from this small region is made for every small region, Claims 1 and 2 creating dynamically subtree which consists of an object contained in this pyramid or a cone, and performing ray tracing using this subtree, or a ray-tracing method given in 3.

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a principle figure of the ray-tracing method of this invention.
[Drawing 2]It is a process flow figure of one example of this invention.
[Drawing 3]It is a process flow figure of a continuation of this invention.
[Drawing 4]It is a figure showing division of an image face (screen).
[Drawing 5]It is a figure showing the general example of the tree structure of object definition data.
[Drawing 6]It is a figure showing the project area of the bounding volume (BV) of object space, and the relation of intersection judging cost.
[Drawing 7]It is a figure showing optimization of subtree.
[Drawing 8]It is an explanatory view of sorting of subtree.
[Drawing 9]It is a figure showing the example of a test of the uniformity of a small region.
[Drawing 10]It is a figure showing the example of a test of the uniformity of a small region similarly.
[Drawing 11]It is an explanatory view of paraxial light flux.
[Drawing 12]It is a figure showing the example of composition of the bounding volume (PBV) of light flux.
[Drawing 13]It is a figure showing the example of improvement of PBV.
[Drawing 14]It is an example of image generation.
[Drawing 15]It is an example of image generation.
[Drawing 16]It is an example of image generation.
[Drawing 17]It is an example of image generation.
[Drawing 18]It is an example of image generation.
[Description of Notations]
1 Viewpoint (eyes)

2 Image face (screen)
2' Small region
3 Object body space
4 Beam of light
4' Reflected ray
BV Bounding volume
P Object (primitive)

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application]This invention starts the field of the image generation by the ray-tracing method, and relates to the high speed processing method of the ray tracing method in detail.
[0002]
[Description of the Prior Art]If the image generation method by the ray-tracing method pursues a beam of light toward the direction of objective from a viewpoint and a beam of light intersects an object, it will be the technique of projecting on an image face (screen) in quest of the luminosity of the object in the point, and generating a picture, but the problem is that calculation takes most time. The greater part of this computation time is spent on the intersection judging of a beam of light and an object. Therefore, the proposal of the data structure and algorithm which aimed at reduction of the number of times of an intersection judging is performed.
[0003]There is essence of improvement in the speed of a ray-tracing algorithm in asking for the object projected on the pixel efficiently from each pixel of an image face, if it attaches. Various high speed techniques have been proposed from before for the purpose of it. If the conventional technique is classified, it will roughly be divided into use of the formation of ** data hierarchy, and Boundind Volume (BV), and use of **Ray coherency.
[0004]The method of aiming at reduction of the number of times of an intersection judging by hierarchization of object definition data and use of BV is known well. The hierarchization using Octree or a ConstructiveSolid Geometry (CSG) model is the example. Necessity is accepted to each node of the tree hierarchized in these models in BV which includes all object RIMITIBU of the leaf node which serves as posterity of the node, and it is setting-out ****. The number of times of intersection calculation is substantially reducible by performing an intersection judging to these BV(s). Based on the fact "the number of the beams of light which intersect an object,

so that volume of BV is made small decreases", examination which calculates efficiently BV to which volume becomes as small as possible is performed.

[0005]On the other hand, Arvo and others classifies a beam of light according to a viewpoint and a direction, and the grouping of the similar beam of light is carried out, By asking for the candidate of the object which carries out an intersection judging by asking for the candidate of the object which intersects those beams of light. The number of candidates of the object which carries out an intersection judging. The technique to reduce is proposed (Arvo J., Kirk D.: "Fast Ray Tracing by ray Classification", Comput.Graphics, Vol.21.No.4, pp.55-64, Jul.1987). Since this method can perform judgment of candidate objects efficiently, its high speed effect of the ray tracing method is large.

[0006]

[Problem(s) to be Solved by the Invention]Although any method of the above of the high speed effect is large, it is necessary to realize still more nearly high-speed processing. Since the conventional technique was performing ray tracing, using the hierarchized tree structure as it is, it needs intersection judging calculation of the wooden depth at least, and had become a narrow path of this computational complexity reduction.

[0007]In the method of generating a picture, the purpose of this invention has the high speed effect in providing the very high ray-tracing method, pursuing a beam of light and asking for objective luminosity.

[0008]

[Means for Solving the Problem]To achieve the above objects, in a method of generating a picture, while claim 1 pursues a beam of light toward the direction of objective from a viewpoint and asking for objective luminosity, Subtree which consists of an object projected on this small region is dynamically created for every small region of an image face using object definition data hierarchized by tree structure, and ray tracing is performed using this subtree.

[0009]An invention of claim 2 rearranges dynamically into order with a near distance from a viewpoint bounding volume set as each node of subtree to a beam of light to pursue, and searches for subtree in order with this near distance.

[0010]An invention of claim 3 omits search of subtree in this small region, when an object projected on a small region judges whether it is occupied by the same object and occupied by the same object.

[0011]About reflection and a refracted ray, an invention of claim 4 makes **** or a cone which includes a large majority of a beam of light reflected and refracted from this small region for every small region, creates dynamically subtree which consists of an object contained in this pyramid or a cone, and performs ray tracing using this subtree.

[0012]

[Function]In this invention, by making subtree from the object definition data of the given tree

structure dynamically, and making the wooden depth shallow, the number of objects projected on the small region of an image face (screen) decreases, and can reduce the number of times of an intersection judging. By adding the uniformity test of the object in sorting by the distance from the viewpoint in each node level of subtree, and a small region, narrowing down of the candidate objects which intersect a beam of light can be performed very efficiently, and high speed processing becomes possible further. About reflection and the refracted light, not only a primary beam of light but the picture of ray tracing by reflection and the refracted light is simultaneously generated by making subtree further for every small region, and performing ray tracing using this subtree.

[0013]

[Example]Hereafter, a drawing explains one example of this invention.

[0014]Drawing 1 (a) shows the principle figure of the ray-tracing method of this invention. As for eyes (viewpoint) and 2, in drawing 1, object space and 4 are beams of light an image face (screen) and 3 1. Many objects (primitive) Pi are on the object space 3 for image generation. This object space 3 is divided hierarchical until the primitive contained in each subspace will be one piece, the bounding volume (BV) which includes all the primitives included in that space is made to each subspace, and objective definition data is hierarchized. The layered structure of the object definition data corresponding to the object space 3 of drawing 1 (a) is shown in drawing 1 (b). Here, a round mark is a node, the node BVi corresponding to bounding volume (BV) expresses the coordinates of the BV concerned, etc., and the node Pi corresponding to an object (primitive) expresses the equation of the object concerned, a surface attribute, a center position, etc. Hierarchization of this object definition data is mentioned later.

[0015]In this invention, the image face (screen) 2 is divided into much small region 2', an intersection judging is performed for every small region with reference to the layered structure of the object definition data given beforehand, and the subtree which consists of an object projected on this small region is created dynamically. And ray tracing is performed using this subtree, and it asks for the luminosity of the object projected on this small region for every small region, and plots to the image face 2 (rendering). If there is reflected ray (refracted ray) 4' at this time, the subtree to it is also generated dynamically, this subtree will be used, ray tracing will be performed further, and the picture of a reflective (refraction) object will also be plotted in piles to an applicable small region. The above processing is repeated and carried out about each small region.

[0016]The flow chart of one example of the image generation processing by the ray-tracing method of this invention is shown in drawing 2 and drawing 3, and the explanatory view which divides an image face (screen) into a small region is shown in drawing 4.

[0017]Drawing 2 is processing until it performs a beam of light, BV, or the intersection judging with an object (primitive) to every [of the image face 2] small region 2' and creates the

subtree to this small region using the object definition data hierarchized by the tree structure. Suppose horizontally that processing is carried out to the right and a perpendicular direction in lower order from a top from the left about each small region of the image face 2 in this example. Drawing 4 is a portion which performs ray tracing using the created subtree (pursuit of the reflective refracted light is also performed), and performs the primitive rendering projected on it about one small region. This rendering is repeated about each small region, and is performed.
[0018]Processing of drawing 2 and drawing 3 is carried out on a computer actually [all]. Below, the main processings are explained in full detail.
[0019]There are uniform division and un-uniform division as a split plot experiment of the hierarchization space of object definition data. Although which is chosen should synthesize and opt for the cost which makes a data structure, and the cost of ray tracing, since it is generally dependent on data, right and wrong cannot generally be said. Here, the latter division is considered. The latter division serves as a tree structure. Space is specifically divided into k subspaces and an object (primitive) is divided into each subspace. When a primitive straddles subspace, it puts into one of subspaces. For example, volume is put into the subspace shared most greatly. A rate is recursively continued at this rate until the primitive which each subspace contains will be one piece. If division finishes, the bounding volume (BV) which includes all the primitives included in the space will be made to each subspace. The data structure of k **** is made by this division. It is necessary to determine the value of k.
[0020]If the tree balances, it can decide as follows. Suppose that it hierarchized by k **** now as it **(ed) to drawing 5. BV is set to each intermediate node of this tree. If ray tracing is carried out using this tree, BV and the intersection judging which were set as all the child nodes of a root node will be first performed to each beam of light. Subsequently, one node is chosen by a certain method among these nodes, and an intersection judging is performed to all those child nodes. The number of times N of an intersection judging to the beam of light if an intersection judging progresses without backtracking to a leaf node [Equation 1]

$$N = k \cdot \log_k n$$

It becomes. Here, n is a primitive number.
[0021]k which minimizes N when n is fixed is k=e (base of a natural logarithm). Since k is an integer, k= 3 gives the minimum of N actually. Subsequently, it is set to k= 2 and 4. 2 and 4 give the same value. Therefore, it is effective to consider it as a ternary tree or 2 (4) ********. That is, backtracking does not have **, or if a node selection method with very little generating is given, three hierarchization are and according to 2 (4) **** will become advantageous. Below, compatibility with other high speed techniques is considered, and a binary tree is used. Drawing 1 (b) is an easy example of a binary tree.

[0022]Although it is desirable to set up so that volume may become small as much as possible as for the shape of BV, and the intersection judging BV, setting out in consideration of intersection judging cost must be carried out. The tree structure by which BV was set as each node as shown in drawing 6 (a) now is considered, and suppose that the projection to the image face (screen) became drawing 6 (b). Here, a project area is expressed with Si. If a beam of light strikes upon the node n1, the intersection judging with BV of the child node will be performed. Therefore, the time T which the intersection judging with the child node takes under the conditions that the beam of light struck upon n1 is the whole screen, [Equation 2]

$$T = 2 C_1 S_1$$

It becomes. Here, C_1 is intersection judging cost with BV set as n1.

[0023]If there is no backtracking, this formula will be materialized to each node of the tree to search. In this case, if intersection judging cost is low even if it uses simple-shaped BV from this formula, total computation time may decrease. (Computation time is the same, if cost is a half even if a project area doubles.) Below, the rectangular parallelepiped in alignment with an axis (world coordinate) is used as BV.

[0024]Carrying out ray tracing using the tree first given to each pixel of a dynamic subtree image face (screen) has much futility. It is because the primitive number projected on the small region of an image face is restricted. Then, subtree is dynamically made from the given tree for every small region, and it is possible in a small region to carry out ray tracing using subtree. Here, a primary beam of light (beam of light pursued from a viewpoint) is described. As a small region, if square regions or a rectangle region is taken, subtree will be made from the primitive which goes into the pyramid made from a viewpoint and the point of four corners of a field, or intersects a pyramid (the bunch of the beam of light in this pyramid is called light flux). If this production cost is high, it is disadvantageous, but the next consideration shows actually that it is low.

[0025](b) Although a pyramid creation pyramid comprises four flat surfaces, it is not necessary to search for these flat surfaces for every small region. In the pyramid to the small region horizontally located in a line on the image face, a horizontal field (the 2nd page of the upper and lower sides are) turns into the flat surface common to any pyramid, and same (for example, HF_0, HF_1, etc. of drawing 4). In the small region perpendicularly located in a line, the same relation is materialized to a vertical field (VF_0, VF_1, etc. of drawing 4). An adjoining field is sharable. By using half space of a plane normal line direction as the outside, and making an opposite hand into the inside can prescribe the inside of a pyramid to a meaning. Therefore, in the size of an image face, if the size of nxn and a small region is made into mxm, the number of required flat surfaces will be set to 2 (n/m+1) on the whole.

[0026](**) Give four flags to each node of a dynamic subtree creation tree. It is shown whether

each flag corresponds to the field of a pyramid and there is any BV or primitive corresponding to a node inside a field, there is outside, or it crosses. [any] The flag of each node is set up for the first time from the wooden root node. This can be found in the intersection judging of drawing 2. this time -- the inside of the field in a certain node -- or if it becomes outside, flag setting out to the field concerned below that child node will become unnecessary. This is clear from the character which BV has.
[0027]If image generation is subsequently performed in order of a horizontal small region in a small region, the flag to the level surface can be used in common to all the small regions horizontally located in a line. The flag setting out to vertical confrontation can exclude an unnecessary inside-and-outside judging by referring to the flag to the level surface. Drawing 2 shows such processing. Subtree is constituted referring to these flags. That is, the subtree which consists only of BV and the primitive which intersect a pyramid or are contained in a pyramid is constituted. The optimization shown in drawing 7 is also performed at this time. Since the effect of **** works so that clearly from how to make, a subtree production cost can be held down low. The subtree creating method described here is applicable similarly to a secondary beam of light (reflection and refracted light).
[0028]The primitive near [sorting] a viewpoint is projected on an image face, and even if BV which includes the primitive compares with BV which includes other primitives, it performs sorting by the distance from a viewpoint on each level of subtree based on the fact that it is closer to a viewpoint in many cases. That is, a child node will be rearranged into order with a near distance from a viewpoint, if BV and the beam of light of a certain node will cross and it will be necessary to perform an intersection judging to the child node. This is performed for every beam of light.
[0029]If each wooden level has performed sorting when performing tree search of depth direction priority, As shown in drawing 8, when the distance from the primitive viewpoint acquired by searching below for a certain child node is nearer than the distance from the viewpoint of BV set as the following child node, the search below the child node becomes unnecessary. Thus, **** using sorting is made.
[0030]If a binary tree is used, sorting of a child node requires only one comparison, and is very efficient. There is a reason for having used the binary tree here.
[0031]What is necessary is just to perform that primitive rendering in this field, if it turns out by a certain method that the inside of the uniformity test small region in a small region is the same primitive projection image. The effect which means that it becomes unnecessary in a field looking this for a tree, and is given to improvement in the speed is large. It is required that this test should be as simple as possible.
[0032]Here, the easy following testing methods are performed. Ray tracing is carried out to the point of four corners of a small region. Sorting is carried out with the beam of light which

pursued subtree to the 4th in this stage. Now, suppose that the primitive of the point of four corners was the same. It is set to P. What is necessary is just to investigate that other primitives cannot be found from the primitive P in a small region at the place near a viewpoint at this time. Subtree is searched by Left-most Depth-First, and the primitive Q is found. Q is at a wooden leftmost end. The method of wooden composition to Q is closer to a viewpoint than P. If P and Q are not the same, this field will presuppose that it is not uniform. P and Q presuppose that it was the same. The younger brother node of P is investigated at this time. A younger brother node is in a primitive or the paddle gap to which BV is set. It is set to R. R investigates whether the beam of light pursued to the 4th is intersected. This should just see a flag. If it crosses, the inside of a field is uniform. Otherwise, suppose that it is not uniform.
[0033]Although the judgment that this test is certainly uniform when uniform in a field is not given, it is judged with the inside of a field being uniform, and solves, and it is guaranteed that other primitives cannot be found before the primitive P. The example is shown in drawing 9 and drawing 10. Although drawing 9 (example in which it judges that a) is not uniform, and said (b) is originally uniform, it is an example judge that is not uniform. It is an example judge that is [drawing 10] uniform. In the case of drawing 10, there is no another primitive in a primitive front face, but since the portion which is not primitive is in a small region, it will pursue using subtree to the portion. When there is no PURIMIDIBU in four corners, conditions no primitive to be into a field are that subtree is empty.
[0034]When the inside of a field is not uniform, a field is divided into two and a uniformity test is done. In the range which is not less than a threshold, area size repeats this recursively.
[0035]The light flux which passes through the rectangular area of the pursuit screen of reflection and a refracted-light bunch can be expressed with a truncated pyramid, and can apply the technique described so far. However, it is difficult to describe correctly general reflection and refracted-light bunch. Here, ** reflection and a refracted-light bunch are approximated, and it asks for the bounding volume (hereafter referred to as PBV) of the approximate light flux containing it.
** To approximate PBV, ask for subtree and test uniformity.
** When it is contained in approximation PBV calculated by ** to each reflection and refracted ray, or no is investigated and it is contained, use the result obtained by **.
An approach to say is taken. Becoming a problem here is the point how to calculate approximation PBV.
[0036]When **** of light flux is small, Paraxial approximation as approximation of reflection and refraction. The effective thing is known (Shinya.). M. -- Takahashi T., Naito S.: "Principesand Applications of Pencil Tracing", Comput.Graphics, Vol.21, No.4, pp.45-54, and Jul.1987. Then, approximation PBV will be calculated using paraxial approximation.
[0037]According to paraxial theory, the paraxial light flux which emitted one point (for example,

viewpoint) has the two focal lines, and the direction of the focal line intersects perpendicularly. That is, paraxial light flux is described as a field surrounded by the group of two flat surfaces which has a nodal line which intersects perpendicularly mutually, as shown in drawing 11. Therefore, it is rational to use this field as approximation of light flux. However, not ideal approximation light flux but many beams of light will protrude actual light flux outside the field of approximation light flux (drawing 12 (a)). Then, it considers calculating PBV of approximation light flux based on the beam of light of four corners of light flux. Although a majority of this way of asking exists, easy methods include the technique of making the biggest angle ****** of PBV among the deflection angles of the beam of light of four corners. In the two-dimensional example shown in drawing 12 (a), since it is |theta1|>|theta2|, PBV like drawing 12 (b) is obtained. Specifically, approximation beam-of-light PBV is calculated in the following procedures.

[0038]** Ask for reflection and the refracted ray i of the beam of light corresponding to four corners of a screen small region from a Snell rule.

** Take the average of four obtained beams of light, and consider it as an axial beam of light.

** Calculate suitable beam-of-light coordinates x-y-z which sets z-axis as an axial beam of light. However, the starting point is taken so that the thing of most this side may be set to z= 0 among the starting points of four beams of light.

** Change the starting point and the direction of four beams of light into a beam-of-light coordinate system, and ask for intersection r_i= (r_{xi}, r_{yi}) with the field of z= 0, and direction S_i= (S_{ix}, S_{iy}) = (dx/dz) ($_i$, $_i$ (dy/dz)).

** r_{xmax}=max (r_{xi}), r_{xmin}=min (r_{xi}), r_{ymax}=max (r_{yi}), r_{ymin}=min (r_{yi}) and s_{xmax}=max ($|s_{xi}|$), and s_{ymax}=max ($|s_{yi}|$) are calculated.

** Four flat surfaces to search for are given with a following formula.

[Equation 3]

$$x - s_{xmax} * Z = r_{xmax}$$

$$x + s_{xmax} * Z = r_{xmin}$$

$$y - s_{ymax} * Z = r_{ymax}$$

$$y + s_{ymax} * Z = r_{ymin}$$

Although how to take x and the y-axis is arbitrary, it is efficient to take in accordance with the direction of the focal line.

[0039]This approximation PBV creating method is the simplest, and can consider various modification. The following can be considered as the example.

** It is possible not only the beam of light of four corners but to ask for reflection and the

refracted ray of the beam of light corresponding to all the pixels, and to bound these by the same technique. It is guaranteed that all the beams of light which should be pursued by this are contained in PBV.

** In this PBV, when converging after light flux's reflecting, include useless space (drawing 13 (a)). Therefore, it bounds by two Kon bunch light bunches as shown in drawing 13 (b) about convergence luminous flux.

[0040]If a small region is uniform, subtree is made from a point of four corners by PBV which expects a light source, and a shadow is tested. If it is a parallel ray, PBV serves as a square pillar. Since it is not dependent on a place and the **** vector of each field which constitutes a square pillar becomes anywhere fixed, the cost which makes a square pillar is small.

[0041]Next, the following algorithms are prepared and the high speed effect of this invention is shown. The algorithm of A is premised on use of a binary tree structure below.

A1 : Only BV is used.

A2 : BV and sorting are used.

A2': Use subtree and BV.

A3 : subtree, BV, and sorting are used.

A4 : all are used. (A3+1 Mr. nature test)

The technique of Arvo is used as a comparative object.

B1 : Algorithm of Arvo [0042]The example of the effect of improvement in the speed is shown about the following five kinds of data.

1) Arrange the ball of a fixed radius in in space regularly. The ball of two kinds of radii is prepared. (Two kinds)

2) Arrange at random the ball which has a random radius in space. (One kind)

3) One piece and eight teapots (two kinds)

[0043]The example of image generation is shown in drawing 14 thru/or drawing 18. The pixel number of the generated picture is 512x512. The small region is based on the size of 8x8. Drawing 14 thru/or the primitive total of drawing 16 are 512, and it of drawing 17 and drawing 18 is 522-4416, respectively. BV is using the rectangular parallelepiped in alignment with an axis.

[0044]Image generation time and the number of times of an intersection judging are shown in Table 1 and Table 2.

[Table 1]

計算時間

ｱﾙｺﾞﾘｽﾞﾑ	写真1	写真2	写真3	写真4	写真5
A 1	36'50"	20'46"	55'18"	12'19"	29'08"
A 2	14'07"	17'52"	21'18"	8'42"	18'52"
A 2'	5'12"	2'13"	7'16"	3'08"	7'56"
A 3	4'39"	2'17"	6'46"	2'29"	5'19"
A 4	3'45"	2'10"	5'22"	2'16"	5'03"
B 1	6'55"	2'51"	9'34"	2'29"	6'06"

[Table 2]

交差判定回数

アルゴリズム		写真1	写真2	写真3	写真4	写真5
A1	BV	57.9	37.9	86.6	20.4	50.9
	prim	19.8	6.88	28.8	4.61	9.60
	総数	20355	11744	30245	6552	15849
A2	BV	20.2	29.7	30.9	13.5	30.5
	prim	3.75	5.01	7.28	2.53	4.69
	総数	6281	9087	10008	4190	9228
A2'	BV	3.72	0.66	5.68	2.56	8.12
	prim	3.38	1.61	4.39	2.78	7.04
	総数	1859	595	2640	1400	3974
A3	BV	3.16	0.65	5.09	1.86	4.88
	prim	2.27	1.56	3.48	1.60	3.42
	総数	1424	581	2248	908	2174
A4	BV	1.93	0.65	3.18	1.75	4.61
	prim	2.04	1.79	2.99	1.52	3.20
	総数	1029	546	1608	893	2115
B1	prim	7.80	3.10	10.8	2.51	5.79
	総数	2045	813	2829	657	1516

BVとprimの項は光線1本あたりの交差判定回数（全光線に対する平均）
総数の項は画像面全体での総交差判定回数（BV＋プリミティブ）（×10³）

[0045]This result shows the high speed effect of the algorithm of this invention as follows.
In (a) , algorithm A4 generates a picture most any data for a short time. Even if it compares B1 with A3, there is less image generation time of A3 than B1.
(b) The effect which subtree contributes to improvement in the speed from comparison of A1 and A2' is very high. This is an effect acquired from the intersection decision circuits of A2' decreasing in number remarkably, as shown in Table 2.
(c) Depend for the effect of the uniformity test of a field on a primitive size. Like drawing 15, even when there are few uniformity fields, there is the high speed effect.
(d) In A2, as it turns out that BV is compared with a primitive intersection decision circuit, there

is very little backtracking at the time of tree search, and by using sorting shows things.
[0046]
[Effect of the Invention]As explained above, in the invention of claim 1, by making ****** from the object definition data of the given tree structure dynamically, and making the wooden depth shallow, the number of objects projected on the small region of an image face (screen) decreases, and can reduce the number of times of an intersection judging.
[0047]In the invention of claims 2 and 3, by adding the uniformity test of the object in sorting and a small region with the distance from the viewpoint in each node level of subtree, narrowing down of the candidate objects which intersect a beam of light can be performed very efficiently, and high speed processing becomes possible further.
[0048]In the invention of claim 4, not only a primary beam of light but the picture of ray tracing by reflection and the refracted light is simultaneously generated about reflection and the refracted light by making subtree further for every small region, and performing ray tracing using this subtree.

[Translation done.]

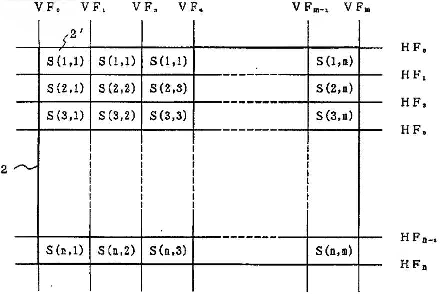
* NOTICES *

JPO and INPIT are not responsible for any
damages caused by the use of this translation.

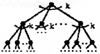
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

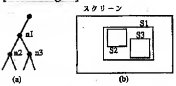
[Drawing 4]



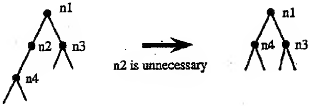
[Drawing 5]



[Drawing 6]

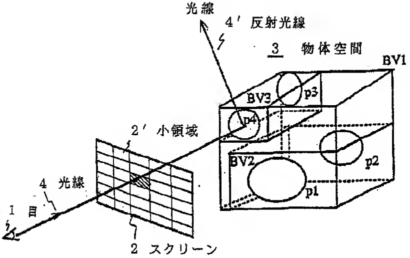


[Drawing 7]



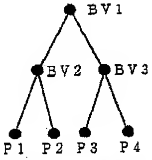
[Drawing 1]

(a)



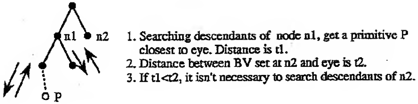
本発明の光線追跡原理図

(b)



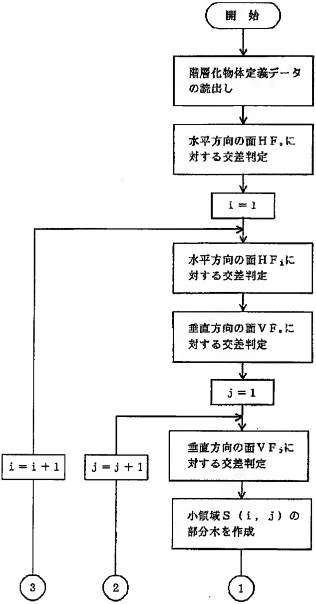
物体の階層化構造

[Drawing 8]

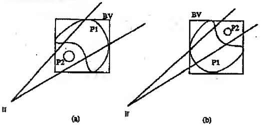


[Drawing 18]

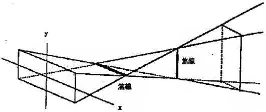
[Drawing 2]



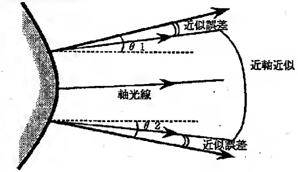
[Drawing 9]



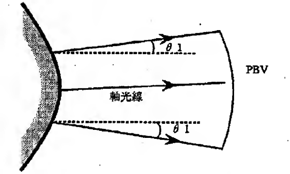
[Drawing 11]



[Drawing 12]

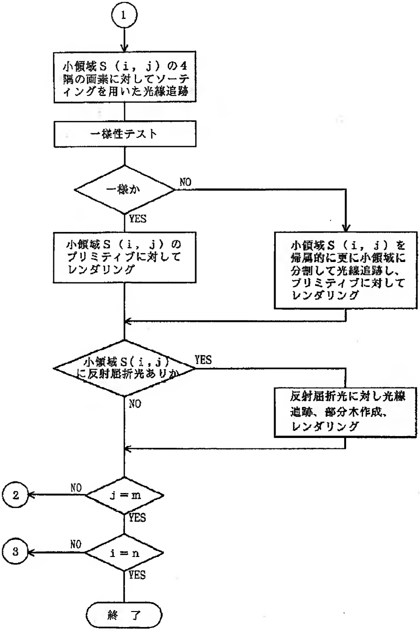


(a)

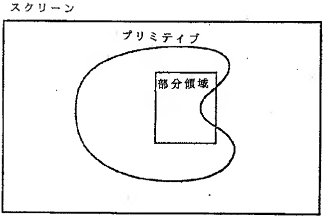


(b)

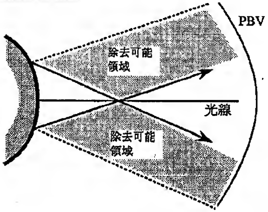
[Drawing 3]


1
小領域S（i，j）の4隅の画素に対してソーティングを用いた光線追跡
一様性テスト
一様か
NO
YES
小領域S（i，j）のプリミティブに対してレンダリング
小領域S（i，j）を帰属的に更に小領域に分割して光線追跡し、プリミティブに対してレンダリング
小領域S(i，j)に反射屈折光ありか
YES
NO
反射屈折光に対し光線追跡、部分木作成、レンダリング
2
NO
j＝m
YES
3
NO
i＝n
YES
終　了

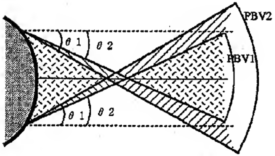
[Drawing 10]



[Drawing 13]



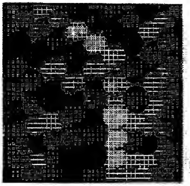
(a)



(b)

[Drawing 14]

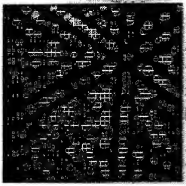
図面代用写真



写 真

[Drawing 15]

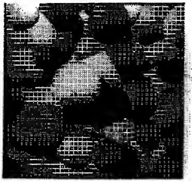
図面代用写真



写 真

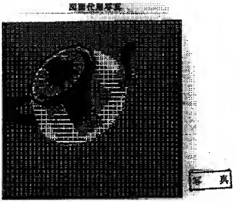
[Drawing 16]

図面代用写真



写 真

[Drawing 17]



[Translation done.]

EFFECT OF THE INVENTION

[Effect of the Invention]As explained above, in the invention of claim 1, by making ****** from the object definition data of the given tree structure dynamically, and making the wooden depth shallow, the number of objects projected on the small region of an image face (screen) decreases, and can reduce the number of times of an intersection judging.
[0047]In the invention of claims 2 and 3, by adding the uniformity test of the object in sorting and a small region with the distance from the viewpoint in each node level of subtree, narrowing down of the candidate objects which intersect a beam of light can be performed very efficiently, and high speed processing becomes possible further.
[0048]In the invention of claim 4, not only a primary beam of light but the picture of ray tracing by reflection and the refracted light is simultaneously generated about reflection and the refracted light by making subtree further for every small region, and performing ray tracing using this subtree.

[Translation done.]

* NOTICES *

EXAMPLE

[Example]Hereafter, a drawing explains one example of this invention.
[0014]Drawing 1 (a) shows the principle figure of the ray-tracing method of this invention. As for eyes (viewpoint) and 2, in drawing 1, object space and 4 are beams of light an image face (screen) and 3 1. Many objects (primitive) Pi are on the object space 3 for image generation. This object space 3 is divided hierarchical until the primitive contained in each subspace will be one piece, the bounding volume (BV) which includes all the primitives included in that space is made to each subspace, and objective definition data is hierarchized. The layered structure of the object definition data corresponding to the object space 3 of drawing 1 (a) is shown in drawing 1 (b). Here, a round mark is a node, the node BVi corresponding to bounding volume (BV) expresses the coordinates of the BV concerned, etc., and the node Pi corresponding to an object (primitive) expresses the equation of the object concerned, a surface attribute, a center position, etc. Hierarchization of this object definition data is mentioned later.
[0015]In this invention, the image face (screen) 2 is divided into much small region 2', an intersection judging is performed for every small region with reference to the layered structure of the object definition data given beforehand, and the subtree which consists of an object projected on this small region is created dynamically. And ray tracing is performed using this subtree, and it asks for the luminosity of the object projected on this small region for every small region, and plots to the image face 2 (rendering). If there is reflected ray (refracted ray) 4' at this time, the subtree to it is also generated dynamically, this subtree will be used, ray tracing will be performed further, and the picture of a reflective (refraction) object will also be plotted in piles to an applicable small region. The above processing is repeated and carried out about each small region.
[0016]The flow chart of one example of the image generation processing by the ray-tracing method of this invention is shown in drawing 2 and drawing 3, and the explanatory view which divides an image face (screen) into a small region is shown in drawing 4.

[0017]Drawing 2 is processing until it performs a beam of light, BV, or the intersection judging with an object (primitive) to every [of the image face 2] small region 2' and creates the subtree to this small region using the object definition data hierarchized by the tree structure. Suppose horizontally that processing is carried out to the right and a perpendicular direction in lower order from a top from the left about each small region of the image face 2 in this example. Drawing 4 is a portion which performs ray tracing using the created subtree (pursuit of the reflective refracted light is also performed), and performs the primitive rendering projected on it about one small region. This rendering is repeated about each small region, and is performed.
[0018]Processing of drawing 2 and drawing 3 is carried out on a computer actually [all]. Below, the main processings are explained in full detail.
[0019]There are uniform division and un-uniform division as a split plot experiment of the hierarchization space of object definition data. Although which is chosen should synthesize and opt for the cost which makes a data structure, and the cost of ray tracing, since it is generally dependent on data, right and wrong cannot generally be said. Here, the latter division is considered. The latter division serves as a tree structure. Space is specifically divided into k subspaces and an object (primitive) is divided into each subspace. When a primitive straddles subspace, it puts into one of subspaces. For example, volume is put into the subspace shared most greatly. A rate is recursively continued at this rate until the primitive which each subspace contains will be one piece. If division finishes, the bounding volume (BV) which includes all the primitives included in the space will be made to each subspace. The data structure of k **** is made by this division. It is necessary to determine the value of k.
[0020]If the tree balances, it can decide as follows. Suppose that it hierarchized by k **** now as it **(ed) to drawing 5. BV is set to each intermediate node of this tree. If ray tracing is carried out using this tree, BV and the intersection judging which were set as all the child nodes of a root node will be first performed to each beam of light. Subsequently, one node is chosen by a certain method among these nodes, and an intersection judging is performed to all those child nodes. The number of times N of an intersection judging to the beam of light if an intersection judging progresses without backtracking to a leaf node [Equation 1]
$$N = k \cdot \log_k n$$

It becomes. Here, n is a primitive number.
[0021]k which minimizes N when n is fixed is k=e (base of a natural logarithm). Since k is an integer, k= 3 gives the minimum of N actually. Subsequently, it is set to k= 2 and 4. 2 and 4 give the same value. Therefore, it is effective to consider it as a ternary tree or 2 (4) ********. That is, backtracking does not have **, or if a node selection method with very little generating is given, three hierarchization are and according to 2 (4) **** will become advantageous.

Below, compatibility with other high speed techniques is considered, and a binary tree is used. Drawing 1 (b) is an easy example of a binary tree.
[0022]Although it is desirable to set up so that volume may become small as much as possible as for the shape of BV, and the intersection judging BV, setting out in consideration of intersection judging cost must be carried out. The tree structure by which BV was set as each node as shown in drawing 6 (a) now is considered, and suppose that the projection to the image face (screen) became drawing 6 (b). Here, a project area is expressed with Si. If a beam of light strikes upon the node n1, the intersection judging with BV of the child node will be performed. Therefore, the time T which the intersection judging with the child node takes under the conditions that the beam of light struck upon n1 is the whole screen, [Equation 2]
$$T = 2 C_1 S_1$$

It becomes. Here, C_1 is intersection judging cost with BV set as n1.
[0023]If there is no backtracking, this formula will be materialized to each node of the tree to search. In this case, if intersection judging cost is low even if it uses simple-shaped BV from this formula, total computation time may decrease. (Computation time is the same, if cost is a half even if a project area doubles.) Below, the rectangular parallelepiped in alignment with an axis (world coordinate) is used as BV.
[0024]Carrying out ray tracing using the tree first given to each pixel of a dynamic subtree image face (screen) has much futility. It is because the primitive number projected on the small region of an image face is restricted. Then, subtree is dynamically made from the given tree for every small region, and it is possible in a small region to carry out ray tracing using subtree. Here, a primary beam of light (beam of light pursued from a viewpoint) is described. As a small region, if square regions or a rectangle region is taken, subtree will be made from the primitive which goes into the pyramid made from a viewpoint and the point of four corners of a field, or intersects a pyramid (the bunch of the beam of light in this pyramid is called light flux). If this production cost is high, it is disadvantageous, but the next consideration shows actually that it is low.
[0025](b) Although a pyramid creation pyramid comprises four flat surfaces, it is not necessary to search for these flat surfaces for every small region. In the pyramid to the small region horizontally located in a line on the image face, a horizontal field (the 2nd page of the upper and lower sides are) turns into the flat surface common to any pyramid, and same (for example, HF_0, HF_1, etc. of drawing 4). In the small region perpendicularly located in a line, the same relation is materialized to a vertical field (VF_0, VF_1, etc. of drawing 4). An adjoining field is sharable. By using half space of a plane normal line direction as the outside, and making an opposite hand into the inside can prescribe the inside of a pyramid to a meaning. Therefore, in the size of an image face, if the size of nxn and a small region is made into mxm, the number

of required flat surfaces will be set to 2 (n/m+1) on the whole.
[0026](**) Give four flags to each node of a dynamic subtree creation tree. It is shown whether each flag corresponds to the field of a pyramid and there is any BV or primitive corresponding to a node inside a field, there is outside, or it crosses. [any] The flag of each node is set up for the first time from the wooden root node. This can be found in the intersection judging of drawing 2. this time -- the inside of the field in a certain node -- or if it becomes outside, flag setting out to the field concerned below that child node will become unnecessary. This is clear from the character which BV has.
[0027]If image generation is subsequently performed in order of a horizontal small region in a small region, the flag to the level surface can be used in common to all the small regions horizontally located in a line. The flag setting out to vertical confrontation can exclude an unnecessary inside-and-outside judging by referring to the flag to the level surface. Drawing 2 shows such processing. Subtree is constituted referring to these flags. That is, the subtree which consists only of BV and the primitive which intersect a pyramid or are contained in a pyramid is constituted. The optimization shown in drawing 7 is also performed at this time. Since the effect of **** works so that clearly from how to make, a subtree production cost can be held down low. The subtree creating method described here is applicable similarly to a secondary beam of light (reflection and refracted light).
[0028]The primitive near [sorting] a viewpoint is projected on an image face, and even if BV which includes the primitive compares with BV which includes other primitives, it performs sorting by the distance from a viewpoint on each level of subtree based on the fact that it is closer to a viewpoint in many cases. That is, a child node will be rearranged into order with a near distance from a viewpoint, if BV and the beam of light of a certain node will cross and it will be necessary to perform an intersection judging to the child node. This is performed for every beam of light.
[0029]If each wooden level has performed sorting when performing tree search of depth direction priority, As shown in drawing 8, when the distance from the primitive viewpoint acquired by searching below for a certain child node is nearer than the distance from the viewpoint of BV set as the following child node, the search below the child node becomes unnecessary. Thus, **** using sorting is made.
[0030]If a binary tree is used, sorting of a child node requires only one comparison, and is very efficient. There is a reason for having used the binary tree here.
[0031]What is necessary is just to perform that primitive rendering in this field, if it turns out by a certain method that the inside of the uniformity test small region in a small region is the same primitive projection image. The effect which means that it becomes unnecessary in a field looking this for a tree, and is given to improvement in the speed is large. It is required that this test should be as simple as possible.

[0032]Here, the easy following testing methods are performed. Ray tracing is carried out to the point of four corners of a small region. Sorting is carried out with the beam of light which pursued subtree to the 4th in this stage. Now, suppose that the primitive of the point of four corners was the same. It is set to P. What is necessary is just to investigate that other primitives cannot be found from the primitive P in a small region at the place near a viewpoint at this time. Subtree is searched by Left-most Depth-First, and the primitive Q is found. Q is at a wooden leftmost end. The method of wooden composition to Q is closer to a viewpoint than P. If P and Q are not the same, this field will presuppose that it is not uniform. P and Q presuppose that it was the same. The younger brother node of P is investigated at this time. A younger brother node is in a primitive or the paddle gap to which BV is set. It is set to R. R investigates whether the beam of light pursued to the 4th is intersected. This should just see a flag. If it crosses, the inside of a field is uniform. Otherwise, suppose that it is not uniform.
[0033]Although the judgment that this test is certainly uniform when uniform in a field is not given, it is judged with the inside of a field being uniform, and solves, and it is guaranteed that other primitives cannot be found before the primitive P. The example is shown in drawing 9 and drawing 10. Although drawing 9 (example in which it judges that a) is not uniform, and said (b) is originally uniform, it is an example judge that is not uniform. It is an example judge that is [drawing 10] uniform. In the case of drawing 10, there is no another primitive in a primitive front face, but since the portion which is not primitive is in a small region, it will pursue using subtree to the portion. When there is no PURIMIDIBU in four corners, conditions no primitive to be into a field are that subtree is empty.
[0034]When the inside of a field is not uniform, a field is divided into two and a uniformity test is done. In the range which is not less than a threshold, area size repeats this recursively.
[0035]The light flux which passes through the rectangular area of the pursuit screen of reflection and a refracted-light bunch can be expressed with a truncated pyramid, and can apply the technique described so far. However, it is difficult to describe correctly general reflection and refracted-light bunch. Here, ** reflection and a refracted-light bunch are approximated, and it asks for the bounding volume (hereafter referred to as PBV) of the approximate light flux containing it.
** To approximate PBV, ask for subtree and test uniformity.
** When it is contained in approximation PBV calculated by ** to each reflection and refracted ray, or no is investigated and it is contained, use the result obtained by **.
An approach to say is taken. Becoming a problem here is the point how to calculate approximation PBV.
[0036]When **** of light flux is small, Paraxial approximation as approximation of reflection and refraction. The effective thing is known (Shinya.). M. -- Takahashi T., Naito S.: "Principesand Applications of Pencil Tracing", Comput.Graphics, Vol.21, No.4, pp.45-54, and Jul.1987. Then,

approximation PBV will be calculated using paraxial approximation.
[0037]According to paraxial theory, the paraxial light flux which emitted one point (for example, viewpoint) has the two focal lines, and the direction of the focal line intersects perpendicularly. That is, paraxial light flux is described as a field surrounded by the group of two flat surfaces which has a nodal line which intersects perpendicularly mutually, as shown in drawing 11. Therefore, it is rational to use this field as approximation of light flux. However, not ideal approximation light flux but many beams of light will protrude actual light flux outside the field of approximation light flux (drawing 12 (a)). Then, it considers calculating PBV of approximation light flux based on the beam of light of four corners of light flux. Although a majority of this way of asking exists, easy methods include the technique of making the biggest angle ****** of PBV among the deflection angles of the beam of light of four corners. In the two-dimensional example shown in drawing 12 (a), since it is |theta1|>|theta2|, PBV like drawing 12 (b) is obtained. Specifically, approximation beam-of-light PBV is calculated in the following procedures.
[0038]** Ask for reflection and the refracted ray i of the beam of light corresponding to four corners of a screen small region from a Snell rule.
** Take the average of four obtained beams of light, and consider it as an axial beam of light.
** Calculate suitable beam-of-light coordinates x-y-z which sets z-axis as an axial beam of light. However, the starting point is taken so that the thing of most this side may be set to z= 0 among the starting points of four beams of light.
** Change the starting point and the direction of four beams of light into a beam-of-light coordinate system, and ask for intersection $r_i= (r_{xi}, r_{yi})$ with the field of z= 0, and direction $S_i= (S_{ix}, S_{iy}) = (dx/dz) (_{i}, _{i} (dy/dz))$.
** r_{xmax}=max (r_{xi}), r_{xmin}=min (r_{xi}), r_{ymax}=max (r_{yi}), r_{ymin}=min (r_{yi}) and s_{xmax}=max ($|s_{xi}|$), and s_{ymax}=max ($|s_{yi}|$) are calculated.
** Four flat surfaces to search for are given with a following formula.
[Equation 3]

$$x - s_{xmax} * Z = r_{xmax}$$

$$x + s_{xmax} * Z = r_{xmin}$$

$$y - s_{ymax} * Z = r_{ymax}$$

$$y + s_{ymax} * Z = r_{ymin}$$

Although how to take x and the y-axis is arbitrary, it is efficient to take in accordance with the direction of the focal line.
[0039]This approximation PBV creating method is the simplest, and can consider various

modification. The following can be considered as the example.
** It is possible not only the beam of light of four corners but to ask for reflection and the refracted ray of the beam of light corresponding to all the pixels, and to bound these by the same technique. It is guaranteed that all the beams of light which should be pursued by this are contained in PBV.
** In this PBV, when converging after light flux's reflecting, include useless space (drawing 13 (a)). Therefore, it bounds by two Kon bunch light bunches as shown in drawing 13 (b) about convergence luminous flux.
[0040]If a small region is uniform, subtree is made from a point of four corners by PBV which expects a light source, and a shadow is tested. If it is a parallel ray, PBV serves as a square pillar. Since it is not dependent on a place and the **** vector of each field which constitutes a square pillar becomes anywhere fixed, the cost which makes a square pillar is small.
[0041]Next, the following algorithms are prepared and the high speed effect of this invention is shown. The algorithm of A is premised on use of a binary tree structure below.
A1 : Only BV is used.
A2 : BV and sorting are used.
A2': Use subtree and BV.
A3 : subtree, BV, and sorting are used.
A4 : all are used. (A3+1 Mr. nature test)
The technique of Arvo is used as a comparative object.
B1 : Algorithm of Arvo [0042]The example of the effect of improvement in the speed is shown about the following five kinds of data.
1) Arrange the ball of a fixed radius in in space regularly. The ball of two kinds of radii is prepared. (Two kinds)
2) Arrange at random the ball which has a random radius in space. (One kind)
3) One piece and eight teapots (two kinds)
[0043]The example of image generation is shown in drawing 14 thru/or drawing 18. The pixel number of the generated picture is 512x512. The small region is based on the size of 8x8. Drawing 14 thru/or the primitive total of drawing 16 are 512, and it of drawing 17 and drawing 18 is 522-4416, respectively. BV is using the rectangular parallelepiped in alignment with an axis.
[0044]Image generation time and the number of times of an intersection judging are shown in Table 1 and Table 2.
[Table 1]

計算時間

ｱﾙｺﾞﾘｽﾞﾑ	写真1	写真2	写真3	写真4	写真5
A 1	36'50"	20'46"	55'18"	12'19"	29'08"
A 2	14'07"	17'52"	21'18"	8'42"	18'52"
A 2'	5'12"	2'13"	7'16"	3'08"	7'56"
A 3	4'39"	2'17"	6'46"	2'29"	5'19"
A 4	3'45"	2'10"	5'22"	2'16"	5'03"
B 1	6'55"	2'51"	9'34"	2'29"	6'06"

[Table 2]

交差判定回数

アルゴリズム		写真１	写真２	写真３	写真４	写真５
Ａ１	ＢＶ	57.9	37.9	86.6	20.4	50.9
	prim	19.8	6.88	28.8	4.61	9.60
	総数	20355	11744	30245	6552	15849
Ａ２	ＢＶ	20.2	29.7	30.9	13.5	30.5
	prim	3.75	5.01	7.28	2.53	4.69
	総数	6281	9087	10008	4190	9228
Ａ２'	ＢＶ	3.72	0.66	5.68	2.56	8.12
	prim	3.38	1.61	4.39	2.78	7.04
	総数	1859	595	2640	1400	3974
Ａ３	ＢＶ	3.16	0.65	5.09	1.86	4.88
	prim	2.27	1.56	3.48	1.60	3.42
	総数	1424	581	2248	908	2174
Ａ４	ＢＶ	1.93	0.65	3.18	1.75	4.81
	prim	2.04	1.79	2.99	1.52	3.20
	総数	1029	546	1608	893	2115
Ｂ１	prim	7.80	3.10	10.8	2.51	5.79
	総数	2045	813	2829	657	1516

ＢＶとprimの項は光線１本あたりの交差判定回数（全光線に対する平均）
総数の項は画像面全体での総交差判定回数（ＢＶ＋プリミティブ）（$\times 10^3$）

[0045]This result shows the high speed effect of the algorithm of this invention as follows.
In (a) , algorithm A4 generates a picture most any data for a short time. Even if it compares B1 with A3, there is less image generation time of A3 than B1.
(b) The effect which subtree contributes to improvement in the speed from comparison of A1 and A2' is very high. This is an effect acquired from the intersection decision circuits of A2' decreasing in number remarkably, as shown in Table 2.
(c) Depend for the effect of the uniformity test of a field on a primitive size. Like drawing 15, even when there are few uniformity fields, there is the high speed effect.
(d) In A2, as it turns out that BV is compared with a primitive intersection decision circuit, there

is very little backtracking at the time of tree search, and by using sorting shows things.

[Translation done.]

MEANS

[Means for Solving the Problem]To achieve the above objects, in a method of generating a picture, while claim 1 pursues a beam of light toward the direction of objective from a viewpoint and asking for objective luminosity, Subtree which consists of an object projected on this small region is dynamically created for every small region of an image face using object definition data hierarchized by tree structure, and ray tracing is performed using this subtree.
[0009]An invention of claim 2 rearranges dynamically into order with a near distance from a viewpoint bounding volume set as each node of subtree to a beam of light to pursue, and searches for subtree in order with this near distance.
[0010]An invention of claim 3 omits search of subtree in this small region, when an object projected on a small region judges whether it is occupied by the same object and occupied by the same object.
[0011]About reflection and a refracted ray, an invention of claim 4 makes **** or a cone which includes a large majority of a beam of light reflected and refracted from this small region for every small region, creates dynamically subtree which consists of an object contained in this pyramid or a cone, and performs ray tracing using this subtree.

[Translation done.]

OPERATION

[Function]In this invention, by making subtree from the object definition data of the given tree structure dynamically, and making the wooden depth shallow, the number of objects projected on the small region of an image face (screen) decreases, and can reduce the number of times of an intersection judging. By adding the uniformity test of the object in sorting by the distance from the viewpoint in each node level of subtree, and a small region, narrowing down of the candidate objects which intersect a beam of light can be performed very efficiently, and high speed processing becomes possible further. About reflection and the refracted light, not only a primary beam of light but the picture of ray tracing by reflection and the refracted light is simultaneously generated by making subtree further for every small region, and performing ray tracing using this subtree.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art]If the image generation method by the ray-tracing method pursues a beam of light toward the direction of objective from a viewpoint and a beam of light intersects an object, it will be the technique of projecting on an image face (screen) in quest of the luminosity of the object in the point, and generating a picture, but the problem is that calculation takes most time. The greater part of this computation time is spent on the intersection judging of a beam of light and an object. Therefore, the proposal of the data structure and algorithm which aimed at reduction of the number of times of an intersection judging is performed.
[0003]There is essence of improvement in the speed of a ray-tracing algorithm in asking for the object projected on the pixel efficiently from each pixel of an image face, if it attaches. Various high speed techniques have been proposed from before for the purpose of it. If the conventional technique is classified, it will roughly be divided into use of the formation of ** data hierarchy, and Boundind Volume (BV), and use of **Ray coherency.
[0004]The method of aiming at reduction of the number of times of an intersection judging by hierarchization of object definition data and use of BV is known well. The hierarchization using Octree or a ConstructiveSolid Geometry (CSG) model is the example. Necessity is accepted to each node of the tree hierarchized in these models in BV which includes all object RIMITIBU of the leaf node which serves as posterity of the node, and it is setting-out ****. The number of times of intersection calculation is substantially reducible by performing an intersection judging to these BV(s). Based on the fact "the number of the beams of light which intersect an object, so that volume of BV is made small decreases", examination which calculates efficiently BV to which volume becomes as small as possible is performed.
[0005]On the other hand, Arvo and others classifies a beam of light according to a viewpoint and a direction, and the grouping of the similar beam of light is carried out, By asking for the candidate of the object which carries out an intersection judging by asking for the candidate of

the object which intersects those beams of light. The number of candidates of the object which carries out an intersection judging. The technique to reduce is proposed (Arvo J., Kirk D.: "Fast Ray Tracing by ray Classification", Comput.Graphics, Vol.21.No.4, pp.55-64, Jul.1987). Since this method can perform judgment of candidate objects efficiently, its high speed effect of the ray tracing method is large.

[Translation done.]

TECHNICAL FIELD

[Industrial Application]This invention starts the field of the image generation by the ray-tracing method, and relates to the high speed processing method of the ray tracing method in detail.

[Translation done.]

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention]Although any method of the above of the high speed effect is large, it is necessary to realize still more nearly high-speed processing. Since the conventional technique was performing ray tracing, using the hierarchized tree structure as it is, it needs intersection judging calculation of the wooden depth at least, and had become a narrow path of this computational complexity reduction.
[0007]In the method of generating a picture, the purpose of this invention has the high speed effect in providing the very high ray-tracing method, pursuing a beam of light and asking for objective luminosity.

[Translation done.]